



JOB SHEET 7

Perulangan 2

1. Tujuan

- Mahasiswa memahami konsep perulangan bersarang
- Mahasiswa dapat menjelaskan format penulisan perulangan bersarang (*nested loop*)
- Mahasiswa dapat mengimplementasikan *flowchart* perulangan bersarang menggunakan bahasa pemrograman Java

2. Alat dan Bahan

- PC/Laptop
- JDK
- Java IDE

3. Praktikum

3.1 Percobaan 1: review perulangan yang lalu

1. Percobaan ini ditujukan me-review kembali perulangan yang sudah dibahas pada pertemuan sebelumnya. Pada percobaan 1 akan dibuat program untuk membuat tampilan * sebanyak N kali ke arah **samping**.
2. Buat class baru dengan nama **Star** dan simpan dalam file **Star.java**
3. Buat fungsi/method **main()** di dalamnya.
4. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program.

```
import java.util.Scanner;
```

5. Di dalam fungsi **main()** yang telah dibuat, deklarasikan objek **Scanner** dengan nama **sc**.

```
Scanner sc = new Scanner(System.in);
```

6. Pada baris selanjutnya, tampilkan instruksi untuk memasukan nilai yang akan disimpan ke variabel **N**.

```
System.out.print("Masukkan nilai N = ");  
int N = sc.nextInt();
```



7. Pada baris selanjutnya, buat sintaks perulangan dengan `for` seperti di bawah ini.

```
for(int i=1; i<=N; i++){
    System.out.print("*");
}
```

Catatan: perlu diperhatikan, bahwa yang digunakan adalah perintah `print`, bukan `println` karena kita ingin menampilkan tanpa ada baris baru

8. Compile dan jalan program!
9. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.

```
Masukkan Nilai N = 5
*****
```

Pertanyaan

1. Jika pada perulangan `for`, inialisasi `i=1` diubah menjadi `i=0`, apa yang akibatnya? Mengapa bisa demikian?
2. Jika pada perulangan `for`, kondisi `i <= N` diubah menjadi `i > N`, apa akibatnya? Mengapa bisa demikian?
3. Jika pada perulangan `for`, kondisi step `i++` diubah menjadi `i--` apa akibatnya? Mengapa bisa demikian?

3.2 Percobaan 2 : Bintang Persegi

1. Pada percobaan ke-2 akan dilakukan percobaan tentang *nested loop*. Kasus yang akan diselesaikan adalah untuk membuat tampilan bujursangkar *, dengan panjang sisi sebanyak N. Misalkan N dimasukan 5, maka hasilnya adalah:

```
*****
*****
*****
*****
*****
```

2. Kalau diamati lebih lanjut, sebenarnya mirip dengan kasus percobaan 1 bukan? Jika di percobaan 1, misal input N bernilai 5, maka yang akan dihasilkan adalah ***** (kita bisa anggap ini sebagai *inner loop* yang mencetak 5 bintang *****), maka untuk kasus percobaan 2 ini bukankah hasil dari percobaan 1 tersebut hanya perlu diulang lagi sebanyak N kali? (dengan menambahkan *outer loop* untuk mengulangi proses *inner loop* sebanyak N kali.)
4. Buat class **Square** dan simpan dengan nama file **Square.java**



5. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner. Jadi tambahkan sintaks import di baris atas sendiri program.

```
import java.util.Scanner;
```

6. Buat method **main()**, dan isikan kode program yang sama dengan isi method **main()** di percobaan 1.

```
Scanner sc = new Scanner(System.in);
System.out.print("Masukkan nilai N = ");
int N = sc.nextInt();
for(int i=1; i<=N; i++){
    System.out.print("*");
}
```

7. Compile dan jalankan program! Dan pastikan program jalan seperti saat percobaan 1.
8. Perhatikan sintaks perulangan yang digunakan untuk mencetak * sebanyak N kali ke arah samping. Di step-6 di atas kode **for** (kota merah) kita jadikan sebagai **inner loop**.
9. Kita looping lagi inner loop sebanyak N kali untuk menghasilkan output seperti tahap 1. Maka perlu ditambahkan perulangan luar (**outer loop**).

```
for(int iOuter=1; iOuter<=N; iOuter++){
    for(int i=1; i<=N; i++){
        System.out.print("*");
    }
    System.out.println();
}
```

10. Simpan perubahan, compile dan jalankan program!
11. Amati hasilnya, maka hasilnya harusnya akan serupa dengan tampilan di bawah ini.

```
Masukkan Nilai N = 5
*****
*****
*****
*****
*****
```

Pertanyaan

1. Perhatikan perulangan luar. Jika pada sintaks **for**, inisialisasi **iOuter=1** diubah menjadi **iOuter=0**, apa yang akibatnya? Mengapa bisa demikian?
2. Kembalikan program semula dimana inisialisasi **iOuter=1**. Kemudian perhatikan perulangan dalam, Jika pada sintaks **for**, inisialisasi **i=1** diubah menjadi **i=0**, apa yang akibatnya? Mengapa bisa demikian?



3. Jadi, apakah perbedaan kegunaan antara perulangan luar dengan perulangan yang berada didalamnya?
4. Mengapa perlu ditambahkan sintaks `System.out.println();` di bawah perulangan dalam? Apa akibatnya jika sintaks tersebut dihilangkan?

3.3 Percobaan 3 : Bintang Segitiga

1. Pada percobaan ke-3 akan dilakukan percobaan segitiga *, dengan sama siku dengan tinggi sebesar N. Misalkan N dimasukan 5, maka hasilnya adalah:

```
*
**
***
****
*****
```

2. Buat class **Triangle** dan simpan dengan nama file **Triangle.java**
3. Karena program membutuhkan input dari keyboard, maka perlu import class Scanner.
4. Buat method **main()**, dan isikan kode program berikut kedalam method **main()**.

```
Scanner sc = new Scanner(System.in);
System.out.print("Masukkan nilai N = ");
int N = sc.nextInt();
int i = 0;
while(i <= N) {
    int j = 0;
    while(j < i) {
        System.out.print("*");
        j++;
    }
    i++;
}
```

5. Compile dan jalankan program! Amati apa yang terjadi.

Pertanyaan

1. Perhatikan, apakah output yang dihasilkan dengan nilai N = 5 sesuai dengan tampilan berikut?

```
*
**
***
****
*****
```



2. Jika tidak sesuai, bagian mana saja yang harus diperbaiki/ditambahkan? Jelaskan setiap bagian yang perlu diperbaiki/ditambahkan.

3.4 Percobaan 4 : Kuis Tebak Angka

1. Buat class baru dengan nama Triangle dan simpan dengan nama file **Quiz.java**.

Import class **Random** dan **class Scanner**, di baris awal program.

```
import java.util.Scanner;
import java.util.Random;
```

2. Buat fungsi **main()**
3. Di dalam fungsi **main()** deklarasikan objek dari **class Random** dan **class Scanner**. **Class Random**, pada kasus ini digunakan untuk mengacacak angka.

```
Random random = new Random();
Scanner input = new Scanner(System.in);
```

4. Kemudian pada baris selanjutnya, tambahkan sintaks seperti di bawah ini.

```
char menu='y';
do{
    int number = random.nextInt(10) + 1;
    boolean success = false;
    do {
        System.out.print("Tebak angka (1-10): ");
        int answer = input.nextInt();
        input.nextLine();
        success = (answer == number);
    } while(!success);
    System.out.print("Apakah Anda ingin mengulang permainan (Y/y) ?");
    menu = input.nextLine().charAt(0);
} while (menu=='y' || menu=='Y');
```

Catatan: Statement `input.nextLine()` pada potongan kode di atas, digunakan untuk mengabaikan karakter new line

5. Compile dan jalankan program.
6. Amati jalannya alur program tersebut.

Pertanyaan

1. Jelaskan alur program di atas!
2. Apa yang harus dilakukan untuk tidak melanjutkan (tidak mengulangi) permainan tersebut?



3. Modifikasi program di atas, sehingga bisa menampilkan informasi mengenai : input nilai tebakan yang dimasukan oleh user apakah lebih kecil atau lebih besar dari jawaban/*number* yang di random!

4. Tugas

1. Buatlah program untuk mencetak tampilan segitiga angka seperti di bawah ini berdasarkan input N (nilai N minimal 3). Contoh N = 5

```

1
12
123
1234
12345
    
```

2. Buatlah program untuk mencetak tampilan segitiga bintang seperti di bawah ini berdasarkan input N (nilai N minimal 5). Contoh N = 7

```

*****
*****
*****
****
***
**
*
    
```

3. Buatlah program untuk mencetak tampilan persegi angka seperti di bawah ini berdasarkan input N (nilai N minimal 3). Contoh N = 3, dan N = 5

```

          5 5 5 5 5
          5      5
3 3 3      5      5
3  3      5      5
3 3 3      5 5 5 5 5
    
```

4. Buatlah program untuk mencetak tampilan persegi angka seperti di bawah ini berdasarkan input N (nilai N minimal 5). Contoh N = 5

```

12345
54321
12345
54321
12345
    
```